In the Specification:

Please replace the paragraph beginning on page 4 line 12 with the following amended paragraph:

One approach is to store a pointer, to the page to be written or to the block to be erased, in a predetermined location before the operation, so that when device **10** powers up again, controller **14** can look up this pointer and immediately know which page or block was the last one targeted. This method usually uses one or more validity flags that signal to controller **14** whether the operation completed successfully. See for example co-pending US Published Patent Application No. ~~10/298,094~~ 2003/0099134, which is incorporated by reference for all purposes as if fully set forth herein. That patent application teaches an example of such a method for protecting against power loss during erasing.

Please replace the paragraph beginning on page 5 line 12 with the following amended paragraph:

It should be pointed out that the above discussion applies to the validity of pages as stand-alone entities. It is another question altogether whether the system as a whole is valid even if no page write was interrupted. Such problems can occur, for example, in file systems in which a user-level operation consists of several page-level operations. For example, the creation of a new file involves writing a directory entry, writing one or more sector allocation tables and only then writing the actual file data. If only some of these write operations are completed by the time power fails, while the remaining write operations have yet to begin, then no page is corrupted but the file

system as a whole is corrupted. Methods for protecting against such problems are known (see for example co-pending US Published Patent Application No. ~~10/397,398~~ 2004/0193564) but are beyond the scope of the present invention.

Please replace the paragraph beginning on page 7 line 2 with the following amended paragraph:

In practice, the four possible bit combinations of a two-bit flash cell are stored as four different threshold voltage ranges. In the above example, the threshold voltage ranges are +3.25V to +5.75V for (0,0), +0.75V to +3.25V for (0,1), -1.75V to +0.75V for (1,0) and –4.25V to –1.75V for (1,1). Figure 6 shows these voltage ranges on the abscissa of a histogram of the threshold voltages of a MLC flash memory: because the cells in a flash device are not exactly identical in their characteristics and behavior (due, for example, to small variations in impurities concentrations or to defects in the silicon structure), encoding the same bit pattern in all the cells does not cause all of the cells to have exactly the same threshold voltage. Each peak of the histogram is labeled with the corresponding bit pattern. Because changing either one of the two bits involves changing the same physical attribute (*i.e.*, the threshold voltage) of the cell, it is clear that the process of changing one bit shifts the other bit from its previously stable state. If the change does not complete correctly, it might result in a wrong interpretation for the value of either or both bits.

Please replace the paragraph starting on page 10 line 23 with the following amended paragraph:

In some embodiments of the system of the present invention, the data storage medium and the controller are operationally associated together within a common housing, such as ~~in the case~~ housing **15** of prior art device **10**. In other embodiments of the system of the present invention, the data storage medium and the controller are housed in separate devices and the operational association of the data storage medium and the controller is reversible. For example, the data storage medium may be the flash memory of a flash memory device that is reversibly mounted on a host device, and the controller may be the central processing unit of the host device.

Please replace the paragraph starting on page 12 line 1 with the following amended paragraph:

FIGs. 2 and 3 are schematic illustrations of two different embodiments of a block of a multi-level-cell flash memory[[.]];

Please add the following paragraphs after page 12 line 2:

FIGs. 4A-4D present an example of writing data to the block of FIG. 2 according to the present invention;

FIG. 5 shows an example of a risk zone in a prior art single-bit-cell flash memory architecture;

FIG. 6 is a histogram of threshold voltages in a prior art MLC flash memory.

Please add the following paragraphs after page 15 line 12:

Figures 4A-4D show this process in detail.

Figure 4A shows the initial status of the first four pages **32** of block **30** when the write command is received. The page **32** at address 0 is indicated as "written". The other three pages are "unwritten. The default target of the write command is the page **32** at address 1. This default command places the page **32** at address 0 at risk, as indicated by the shading of the page **32** at address 0.

Figure 4B shows how the present invention writes the new data without putting the data at address 0 at risk. The data at address 0 are copied to the page **32** at address 2. The new data are written to the page **32** at address 3.

Figure 4C shows the status of the first four pages **32** of block **30** after the write operation of the present invention. The pages **32** at addresses 0, 2 and 3 are "written". It does not matter if the data written to the page **32** at address 0 subsequently are corrupted because these data have been copied to the page **32** at address 2. Figure 4D shows the status of the first four pages **32** of block **30** after the optional marking of the pages **32** at addresses 0 and 1 as "deleted".

Please replace the paragraph beginning on page 16 line 5 with the following amended paragraph:

So writing 512 bytes of data to logical page 0 actually distributes the data among (the first 128 bytes of) physical pages 0 of all four flash memories. Suppose that logical pages 0 and 1 have been written, and a power failure occurs during the writing of logical page 3. That power failure puts at risk the data written to logical

pages 0 and 1. Therefore, the risk zone of logical page *m* is the other three logical pages between logical page *m*-(*m* modulo 4) and logical page *m*-(*m* modulo 4)+3. This is illustrated in Figure 5 with respect to the writing of data to logical page 3. Figure 5 shows four flash memories **40**, each with 512-byte physical pages **42**. The first two physical pages **42** and **44** of each flash memory **40** are shown as having four address zones: bytes 0 through 127, bytes 128 through 255, bytes 256-383 and bytes 384-511. Logical page 0 spans bytes 0 through 127 of all four physical page **42**. Logical page 1 spans bytes 128 through 255 of all four physical pages **42**. Logical page 2 spans bytes 256 through 383 of all four physical pages **42**. Logical page 3 spans bytes 384 through 511 of all four physical pages **42**. Logical page 4 spans bytes 0 through 127 of all four physical pages **44**. Logical page 5 spans bytes 128 through 255 of all four physical pages **44**. Logical page 6 spans bytes 256 through 383 of all four logical pages **44**. Logical page 7 spans bytes 384 through 511 of all four physical pages **44**. Logical pages 0-2 are shaded, as being the risk zone of logical page 3. Writing data to logical page 3 places data previously written to logical pages 0-2 at risk. Note that, in this example, each risk zone includes more pages than the example of Figure 2: or three pages per risk zone rather than one.